ResuMail

Production & Next Steps Guide

Run it, put it online for free, send safely, and grow it.

Version 1.1 · November 2026

What this guide covers

ResuMail is the website you now have in `d:\Agents\MailingAgent`. It reads your resume, writes a short professional introduction with simple rules (no AI), and emails that introduction and your resume to every address on your list, one personal email per person.

This guide explains how it works, how to run it on your computer, how to put it on the internet for free, how to send without landing in spam, and what to build next. Every tool and service mentioned has a free option.

Contents

1. How ResuMail works 2

2. Run it on your computer 3

3. Set up the sender email 4

4. Put it online for free (production) 5

5. Send safely and stay out of spam 6

6. Security: what is protected 7

7. Troubleshooting 7

8. Production checklist 8

9. Next steps: ideas to grow ResuMail 8

1. How ResuMail works

ResuMail has two halves. The **website** (HTML, CSS, JavaScript) runs in your browser and does all of the thinking. A small **server** (Node.js) does exactly one job: it signs in to your email account and hands each email to it. Nothing is stored on the server.

Part	Where it runs	What it does
Resume analyzer	Your browser	Reads PDF/DOCX/TXT text and detects name, contacts, ~240 skills, years of experience, degree, certifications. Scores you against 29 role profiles and writes your intro from templates.
Recipient parser	Your browser	Reads Excel/CSV/TXT or pasted text. Finds emails, name, company and role columns, removes duplicates and invalid addresses, and guesses the company from the email domain.
Mail merge	Your browser	Fills {{first_name}}, {{company}}, {{role}}, {{intro}}, {{highlights}} (a quick snapshot for HR: experience, skills, education, achievement, notice period) and more, separately for every recipient.
Send queue	Your browser	Sends one email at a time with a gap, retries temporary errors, pauses/stops, remembers who already got the email.
API server	Node.js (your PC or Vercel)	/api/verify checks your login. /api/send sends one email with your resume attached through SMTP.
Your mail provider	Gmail / Brevo / Zoho ...	Actually delivers the email. Replies come back to your own inbox.

Why one email per person? Recruiters see only their own address, the greeting uses their name, and if one address bounces it does not affect anyone else. This is also how email providers expect personal outreach to look.

Project files

```
server.js            Express server (local computer, or any Node host)
api/*.js             Vercel serverless functions (reuse lib/handlers.js)
lib/handlers.js      The whole backend: verify + send + safety checks
public/index.html    The page
public/styles.css    Design (teal theme, light + dark mode)
public/app.js        Steps, uploads, preview, send queue
public/analyzer.js   Rule-based resume analyzer
public/merge.js      Recipient parser + {{placeholder}} mail merge
public/vendor/       pdf.js, mammoth (DOCX), SheetJS (Excel), served locally
vercel.json          Vercel deployment settings
.env.example         Optional server-side sender settings
```

2. Run it on your computer

This is the quickest way to use ResuMail, and the most private. Everything stays on your PC.

1. Install **Node.js 20 or newer** from nodejs.org (the "LTS" button). You already have Node 26.
2. Open a terminal in the project folder and install the two dependencies (Express and Nodemailer):

```
cd d:\Agents\MailingAgent
npm install
npm start
```

You will see `ResuMail is running at http://localhost:3000`. Open that address in Chrome or Edge. Keep the terminal open while sending; press **Ctrl + C** to stop the server.

> **Tip.** Use `npm run dev` while you edit the code. It restarts the server automatically when a file changes.

Using the app: the 4 steps

Step	What you do	Good to know
1. Sender email	Pick your provider, type your name, email and app password, click **Verify & continue**.	The app tests the login before anything is sent. Tick "Remember" only on your own computer.
2. Resume & intro	Drop your resume. Choose **Analyze my resume** (auto) or **I'll write it myself**.	Auto mode fills every detail (contact, experience, current company, degree, achievement, skills) and every field stays editable. The intro rewrites itself as you edit; pick **Short** or **Detailed**. Manual mode asks only for contact details, the role and your own intro (a guided template helps).
3. Recipients	Drop an Excel/CSV/TXT file or paste emails.	Columns named Name, Email, Company, Role/Position are detected. A sample file is linked on the page.
4. Compose & send	Check the preview, click **Send test to myself**, then **Start sending**.	Pause/Stop anytime. "Download report" gives a CSV of who was sent, failed or pending.

3. Set up the sender email

ResuMail sends through your own email account using SMTP, the standard way programs send email. For safety, providers require an **app password**: a separate 16-character password that only works for sending mail and can be revoked at any time without changing your real password.

Gmail (recommended for most people)

1. Go to myaccount.google.com → **Security** → turn on **2-Step Verification**. App passwords only exist once this is on.
2. Open myaccount.google.com/apppasswords.
3. Type a name such as **ResuMail** and click **Create**.
4. Copy the 16-letter code (spaces don't matter) and paste it in step 1 of ResuMail with your Gmail address.

> **Important.** If the app password page says "not available", your account is a work/school account with app passwords disabled by the admin, or 2-Step Verification is not on. Use a personal Gmail, or use Brevo below.

Other providers

Provider	Free limit (approx.)	What to enter
Gmail	~500 emails/day	Gmail address + app password.
Brevo SMTP	300 emails/day free	Sign up at brevo.com, add and verify your sender email, then go to **SMTP & API**: use the SMTP login and SMTP key, plus your verified "From" email.
Zoho Mail	Free plan available	Zoho address + app password (Zoho → Security → App Passwords). Use the ".in" option for Indian accounts.
Yahoo Mail	~500/day	Yahoo address + app password (Account security → Generate app password).
Other SMTP	Depends	Host, port (465 or 587), username, password and From email from your provider.

Limits change over time and depend on your account's age and history; treat them as upper bounds, not targets.

Optional: a sender stored on the server

If you deploy ResuMail for yourself and don't want to type the password on every device, put it in server environment variables instead. The page then shows a **Server sender** option that only works with your secret access key, so strangers who find your URL cannot send from your account.

```
SMTP_HOST=smtp.gmail.com
SMTP_PORT=465
SMTP_USER=you@gmail.com
SMTP_PASS=your16letterapppassword
SMTP_FROM_NAME=Your Name
ACCESS_KEY=a-long-random-secret-only-you-know
```

Locally, copy `.env.example` to `.env` and fill it in. On Vercel, add the same keys under Project → Settings → Environment Variables. Never commit `.env` to GitHub; it is already in `.gitignore`.

4. Put it online for free (production)

"Production" means the site runs on the internet 24/7 with HTTPS, so you can use it from any device (or share it). The recommended free host is **Vercel**.

> **Important.** Not every free host can send email. **Vercel** allows outgoing SMTP on ports 465 and 587 (only port 25 is blocked), which is what ResuMail uses. **Render's free tier has blocked SMTP ports 25, 465 and 587 since September 2025**, and Railway blocks SMTP on its lower plans, so the "Verify" step would time out there.

Option A: Vercel (recommended, $0)

1. Create free accounts at github.com and vercel.com (sign in to Vercel with GitHub).
2. Put the project on GitHub. In the project folder run the commands below, then create an empty repository called **resumail** on GitHub and push to it.

```
git init
git add .
git commit -m "ResuMail v1"
git branch -M main
git remote add origin https://github.com/<your-username>/resumail.git
git push -u origin main
```

1. In Vercel click **Add New** → **Project**, pick the **resumail** repository and click **Import**.
2. Leave the settings as they are (vercel.json already sets the Framework to "Other" and the output folder to `public`). Click **Deploy**.
3. After about a minute you get a URL such as `resumail-yourname.vercel.app` with HTTPS. Open it, and go through the 4 steps.
4. Every time you `git push`, Vercel redeploys automatically.

Alternative without GitHub: install the CLI with `npm i -g vercel`, then run `vercel` and later `vercel --prod` in the project folder.

Option B: keep it on your own computer

Perfectly fine for personal use: `npm start` whenever you want to send. To reach it from your phone on the same Wi-Fi, open `http://<your-PC-IP>:3000`.

Option C: other Node hosts

`server.js` runs on any Node host (`npm install`, then `npm start`, with the `PORT` variable respected). Before choosing one, check that it allows outgoing SMTP on port 465 or 587. A paid Render instance (about $7/month) does, for example.

Custom domain (optional)

The `.vercel.app` address is free forever. If you want `resumail.yourname.com`, buy a domain (~$10/year; this is the only thing in the guide that costs money) and add it in Vercel → Project → Settings → Domains.

5. Send safely and stay out of spam

Delivery depends on how you send even more than on the software. These habits keep your emails in the inbox and your account in good standing.

- **Start small.** Send a test to yourself first. In the first week, send 20–50 a day, then grow to 50–100. A brand-new Gmail sending 400 identical emails on day one looks like spam.
- **Keep a gap.** Use 5–10 seconds between emails (the "Gap between emails" setting). ResuMail adds a small random variation automatically.
- **Personalise.** Include a name and company column in your sheet so that each email says "Hi Priya … at Freshworks". Personal emails get replies; generic ones get filtered.
- **Use real, relevant contacts only.** Send to recruiters and hiring managers who publish their email for applications. Never buy email lists; high bounce rates get accounts suspended.
- **Plain and professional.** Short subject without ALL CAPS or "!!!", 120–200 words, one attachment (PDF preferred, under 1 MB), and a real signature with your phone and LinkedIn.
- **Respect replies.** If someone asks you not to email again, remove them from your sheet so a future campaign doesn't reach them.
- **Check the report.** After each run, download the CSV report. Remove addresses that failed with "rejected" so your list stays clean.

If you use your own domain later

Sending from you@`yourname.com` through Brevo or Zoho? Add the **SPF**, **DKIM** and **DMARC** DNS records the provider shows you. Gmail and Yahoo now require proper authentication, and without these records your mail is likely to go to spam. Sending from a normal @gmail.com address through Gmail is already authenticated.

6. Security: what is protected

Concern	How ResuMail handles it
Your resume	Read and analyzed only in your browser. It is sent to the server only as the email attachment and is never saved there.
Your app password	Kept in your browser memory (or local storage only if you tick "Remember"). Sent over HTTPS to the server, used to sign in to SMTP, never written to disk or logs. You can revoke it anytime in your Google account.
Strangers using your site	Visitors must supply their own email login. The optional server sender needs your ACCESS_KEY (checked in constant time).
Abuse / spam floods	Per-IP rate limits (40 sends/min, 15 verifies/min), one recipient per request, 2.5 MB attachment cap, resume file types only.
Header / HTML injection	Subjects and names are stripped of line breaks; the email body is HTML-escaped before links are added.
Server-side request forgery	Custom SMTP hosts are checked so they can't point at private/internal networks, and only mail ports (25/465/587/2525) are allowed.
Dependencies	Only two server packages (Express 5, Nodemailer 10) with 0 known vulnerabilities at build time. Run `npm audit` monthly.

7. Troubleshooting

Message you see	What it means / fix
Sign-in failed. For Gmail, use a 16-character App Password…	You used your normal password, made a typo, or 2-Step Verification is off. Create a new app password and paste it.
Could not connect to the mail server	Your internet is down, a firewall blocks port 465/587, or the host blocks SMTP (e.g. Render free tier). Try another network or host Vercel.
Daily sending limit was reached	Your provider's 24-hour quota is used up. Wait a day; pending emails are kept, so just press Start again tomorrow.
The mail server rejected this recipient address	That address doesn't exist. It is marked Failed; remove it from your list.
3 emails in a row failed … Sending stopped	A safety stop. Read the error in the Status column (hover it), fix the cause, then click "Retry failed".
We couldn't read text from this file	The PDF is a scanned image. It is still attached; write the intro yourself, or export your resume again as a text PDF from Word/Google Docs.
The ResuMail server is not reachable	You opened index.html directly or the server isn't running. Run `npm start` and open http://localhost:3000.
Emails arrive in Spam	Follow section 5: smaller batches, personal greeting, plain subject, clean list. Ask a friend to mark the test as "Not spam".

8. Production checklist

	Before you share or rely on it
☐	Run locally, complete all 4 steps, and receive the test email in your own inbox.
☐	Deploy to Vercel and repeat the test from the live URL on your phone.
☐	If you set a server sender: ACCESS_KEY is long and random, and .env is not in GitHub.
☐	Recipient sheet has Name and Company columns, with no bought or scraped lists.
☐	First real run is small (20–50) with a 5–10 second gap.
☐	Download the report after each run and clean failed addresses.
☐	Monthly: `npm audit`, and update with `npm update` if anything is flagged.

9. Next steps: ideas to grow ResuMail

The current version is complete for personal job outreach. These upgrades are listed from easiest to biggest; all can be done on free tiers.

Upgrade	Why	How (free tools)
Multiple templates	Different pitches for startups vs. MNCs, or for referral requests.	Save several subject/body pairs in localStorage and add a dropdown in step 4.
Follow-up emails	A polite follow-up after 5–7 days roughly doubles reply rates.	Use the "sent_at" in the report; add a "Follow up" button that queues a second template for people sent ≥ 5 days ago.
Scheduling	Send at 9–11 am on weekdays when recruiters read mail.	Vercel Cron Jobs (free on Hobby) plus a small database holding the queue.
History across devices	See every campaign from laptop and phone.	Supabase (free Postgres + auth). Store recipients and statuses per user.
User accounts	Let friends use your deployment with their own data.	Supabase Auth or Clerk free tier; keep each user's SMTP password encrypted or prompt each time.
"Sign in with Google" instead of app passwords	Easier setup for non-technical users.	Gmail API with OAuth. Requires a Google Cloud project, and Google verification for public use; Nodemailer supports OAuth2.
Higher volume	Thousands of emails a month.	Brevo / Resend / Amazon SES HTTP APIs with your own domain + SPF/DKIM/DMARC.
Optional AI polish	Rewrite the intro in different tones.	Add a "Polish with AI" button that is off by default, using a free-tier LLM API key the user provides. The rule-based mode stays the default.
Tests & CI	Catch mistakes before deploys.	Move the analyzer/merge tests into `npm test` (node:test) and run them with GitHub Actions on every push.

Costs at a glance

Item	Cost
Code, libraries (Express, Nodemailer, pdf.js, mammoth, SheetJS)	Free, open source
Running on your own computer	Free
Vercel Hobby hosting with HTTPS and .vercel.app domain	Free
Gmail / Brevo / Zoho sending within free limits	Free
Custom domain (optional)	~$10 / year

Good luck with your applications!